

**ADVANCED DISTRIBUTED
SIMULATION TECHNOLOGY II
(ADST II)**

CDRL AB04

**FINAL REPORT
FOR THE
MODSAF PROGRAM**



19981229 023

FOR: STRICOM
12350 Research Parkway
Orlando, FL 32826

BY: Lockheed Martin
ADST-II
P.O. Box 780217

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 074-0188	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing this collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503				
1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE 19 Jun 1996	3. REPORT TYPE AND DATES COVERED Final Report		
4. TITLE AND SUBTITLE Advanced Distributed Simulation Technology II (ADST II) Final Report for the MODSAF Program		5. FUNDING NUMBERS N61339-96-D-0002		
6. AUTHOR(S)				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Lockheed Martin Corporation Lockheed Martin Information Systems ADST II P.O. Box 780217 Orlando, FL 32878-0217		8. PERFORMING ORGANIZATION REPORT NUMBER ADST-II-CDRL-011R-9600166		
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) NAWCTSD/STRICOM 12350 Research Parkway Orlando, FL 32826-3324		10. SPONSORING / MONITORING AGENCY REPORT NUMBER		
11. SUPPLEMENTARY NOTES				
12a. DISTRIBUTION / AVAILABILITY STATEMENT A - Approved for public release; Distribution unlimited			12b. DISTRIBUTION CODE	
13. ABSTRACT (Maximum 200 Words) This Final Report documents the findings of the Mini-Feasibility Analysis Study (FAS) that is currently being performed for the Advanced Distributed Technology Program II (ADST II) ModSAF Life Cycle Support Delivery Order. This Delivery Order (DO) is being executed in two phases, an analysis phase and an execution phase. The first phase, Analysis, is the initial planning task to support the ModSAF program and transition from the ADST I program to the ADST II program. The purpose of the Analysis phase is to begin the planning, preparation, and assumption of duties and responsibilities currently performed by the ADST I contractor to the ADST II team. In addition, the ADST II team will provide support to STRICOM in the formulation of the ADST II ModSAF software development strategy and development of a comprehensive Life Cycle Support Plan that will provide the necessary plans, schedule, tasks, and resources to support the second phase, Execution. The second phase, Execution, will involve the actual performance of those tasks supporting the ModSAF program.				
14. SUBJECT TERMS ADST-II, STRICOM, MODSAF			15. NUMBER OF PAGES	
			16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UL	

ADST-II-CDRL-011R-9600166
19 June 1996

Contract NG1339-96-D-0002

Orlando, FL 32878-0217

19 June 1996

Table of Contents

1. SCOPE.....	1
1.1 BACKGROUND	1
1.2 OVERVIEW OF CURRENT ACTIVITIES AND STATUS TO DATE.....	1
1.2.1 <i>Beta Testing</i>	1
1.2.2 <i>Integration Process Capture</i>	2
1.2.3 <i>Training</i>	3
1.2.4 <i>Meeting Attendance</i>	3
2. PHASE 1 - MINI-FAS TASKS.....	3
2.1 MODSAF TRANSITION PLAN.....	3
2.2 MODSAF STRATEGY.....	4
2.3 MODSAF SOFTWARE DEVELOPMENT PLAN.....	4
2.4 REARCHITECTURE STUDY FOR THE MODSAF LCSC	4
2.5 TECHNICAL SUPPORT	4
2.6 WORK STATEMENT AND ROM.....	4
2.7 PRICING STRATEGY	4
3. PHASE 2 - EXECUTION PHASE OF MODSAF LIFE CYCLE SUPPORT	4

1. SCOPE

This Final Report documents the findings of the Mini-Feasibility Analysis Study (FAS) that is currently being performed for the Advanced Distributed Simulation Technology Program II (ADST II) ModSAF Life Cycle Support Delivery Order.

This Delivery Order (DO) is being executed in two phases, an analysis phase and an execution phase. The first phase, Analysis, is the initial planning task to support the ModSAF program and transition from the ADST I program to the ADST II program. The purpose of the Analysis Phase is to begin the planning, preparation, and assumption of duties and responsibilities currently performed by the ADST I contractor to the ADST II team. In addition, the ADST II team will provide support to STRICOM in the formulation of the ADST II ModSAF software development strategy and development of a comprehensive Life Cycle Support Plan that will provide the necessary plans, schedule, tasks, and resources to support the second phase, Execution. The second phase, Execution, will involve the actual performance of those tasks supporting the ModSAF program.

1.1 Background

ModSAF (Modular Semi-Automated Forces) traces its lineage to the SIMNET program. As its name implies, ModSAF modularizes the software code associated with both SIMNET SAF and ODIN (73 Easting) SAF. In 1993, DARPA began building ModSAF by developing an open architecture, which could be used to create synthetic agents for a variety of Distributed Interactive Simulation (DIS) applications. The initial effort fielded a system in December 1993 to support the What-if Simulation System for Advanced Research and Development (WISSARD) program, which had a requirement for beyond visual range air-to-air engagement scenarios. After the initial release, the remaining Battlefield Operating Systems (BOS) and behaviors were added to ModSAF to fill out the synthetic battlefield. Concurrently, the Simulation, Training, and Instrumentation Command (STRICOM) agreed to fund an effort to document this revised code. In addition, STRICOM is leading the development and under joint funding from ARPA and STRICOM, ModSAF 1.2 was released in June 1994 and included the majority of systems that had been represented in the previous SIMNET SAF version. ModSAF 2.0, was released in October of 1995, and the latest ModSAF version, Version 2.1, was released in May of 1996.

1.2 Overview of Current Activities and Status to Date

Work started on this DO with the execution of a verbal order from STRICOM on March 27, 1996. The current activities and their status to date are described in the following sections.

1.2.1 Beta Testing

Beta testing of ModSAF v2.1 was supported at the Operational Support Facility (OSF) in Orlando, at Ft. Lee, VA and at Loral Federal Systems (LFS), Cambridge, MA. Two ModSAF engineers supported DIS testing with the DIS Testing Suite (DTS) tool at the OSF. Problem trouble reports (PTRs) were generated and submitted on the Configuration Status Accounting (CSA) Web page for errors found during the testing. The results of the testing have been documented in a report and submitted to STRICOM. The content of this report is included as Appendix A of this document.

One ModSAF engineer supported Combat Support Services (CSS) testing at Ft. Lee, VA. Before testing started, the ModSAF engineer conducted a ModSAF Overview course and configured the test

computer to run ModSAF. PTRs were generated and submitted on the CSA Web page for errors found during the testing.

Three ModSAF engineers supported testing at LFS. This testing included running capability tests and the correction of software problems. PTRs were generated and submitted on the CSA Web page for errors found during testing.

1.2.2 Integration Process Capture

While supporting the beta testing of ModSAF 2.1 at LFS, the ModSAF engineers were able to witness and document the ModSAF baseline integration process. Additional steps will be added to the integration process to allow for the review of candidate software before inclusion in a baseline integration, and possibly to support the use of more powerful configuration management tools. The review of candidate software will reduce integration time, and improve the final product, by identifying substandard software early in the integration process. A summary of the steps in the ModSAF baseline integration process follows:

- 1) The version number is changed in order to uniquely identify the current build from all previous builds.
- 2) Merge new source code with the source code from the previous build. This process will likely produce conflicts where the same source files have been independently modified by both the current integration and by previous integrations in a way that precludes an automatic merging of the files.
- 3) Resolve merge conflicts discovered in step 2 by hand editing the conflicting sections of the source files.
- 4) Perform a full build of the ModSAF software.
- 5) Resolve build errors found in step 3. It may be necessary to repeat steps 4 and 5 until an error free build is achieved.
- 6) Run automated test scenarios to ensure that basic ModSAF functionality has not been compromised.
- 7) Perform an optimized software build with compiler options set to produce an executable with maximum performance for the target computer.
- 8) Run the Interdoc and Interck tools to ensure that documentation is complete and is consistent with the software, and that function parameter types and return types are consistent with header file declarations.
- 9) Run TeX to produce on-line hypertext documentation and printable postscript documentation for the texinfo source files.
- 10) Update documentation tree.
- 11) Build the ModSAF software on other target platforms.
- 12) Resolve errors encountered while building for the other target platforms.
- 13) Verify and archive design and SME documentation.
- 14) Perform the capabilities tests provided by the software developer to verify that the integrated functionality is still intact.

19 June 1996

- 15) Resolve problems encountered during the capabilities testing.
- 16) Perform regression testing to ensure that previously integrated functionality has no been compromised by the newly integrated software.
- 17) Resolve any problems encountered during the regression testing.
- 18) Perform performance benchmark testing.
- 19) Resolve any performance problems encountered during benchmark testing.
- 20) Perform profiling.
- 21) Calculate total lines of source code for the integration.
- 22) Generate an export version of the integrated software.
- 23) Generate and distribute an Integration Report.

1.2.3 Training

A three day ModSAF User/Developer Training course was attended by 14 ModSAF engineers at the OSF. A ModSAF Overview course was conducted at Ft. Lee. One ModSAF engineer and two configuration management personnel attended a Clear Case course at Nations.

1.2.4 Meeting Attendance

In support of transition planning and preparation, the following meetings and conferences were supported:

- ALLSAF 2 Conference in Reno, NV during the week of January 8.
- VV&A meeting in Aberdeen, MD during the week of February 5.
- STOW/ModSAF CM coordination meeting at ARL:UT in Austin, TX on February 9.
- ModSAF IPT/IDT meeting in Orlando, FL during the week of February 29.
- ModSAF IPT/IDT meeting in Orlando, FL during the week of April 15.
- DIS Workshop meeting in Orlando, FL during the week of March 11.
- ModSAF IPT/IDT meeting in Orlando, FL during the week of May 6th.

2. Phase 1 - Mini-FAS Tasks

During the Mini-FAS, the following tasks were performed. A task description and status is provided in the following paragraphs.

2.1 *ModSAF Transition Plan*

A ModSAF Transition Plan was developed as part of this contract and was documented as Contract Deliverable Requirement List (CDRL) item AB01. The document tracking number for this CDRL item is ADST-II-CDRL-011R-9600165.

2.2 *ModSAF Strategy*

Based upon discussions with STRICOM, this task as proposed will not be performed during phase 1. The only portion of this task that was performed under this task was the beta testing described in section 1.2.1.

2.3 *ModSAF Software Development Plan*

The ModSAF Software Development was revised as part of this contract and the revision has delivered as Contract Deliverable Requirement List (CDRL) item AB02. The document tracking number for this CDRL item is ADST-II-CDRL-011R-9600167.

2.4 *Rearchitecture Study for the ModSAF LCSC*

A ModSAF Rearchitecture plan has been developed as part of this Mini-FAS and has been documented as Contract Deliverable Requirement List (CDRL) item AB03. The document tracking number for this CDRL item is ADST-II-CDRL-011R-9600207.

2.5 *Technical Support*

The task of providing technical support to STRICOM is currently in progress. Attendance at the STOW 97 meeting at ARL:UT in Austin, Texas was supported under this task during the week of April 22, 1996.

2.6 *Work Statement and Proposal*

The work statement is complete and work has begun on the proposal. The proposal will be completed and submitted to STRICOM by July 30, 1996.

2.7 *Pricing Strategy*

A pricing strategy for ModSAF software development has been completed under this effort and has been included as Appendix B.

3. Phase 2 - Execution Phase of ModSAF Life Cycle Support

The specifics of the following tasks will be provided in the Proposal to be developed in response to the tasking in paragraph 2.6. Projected tasks to be performed under phase 2 include the following:

- a) Integration Support
- b) Release /Acceptance Testing
- c) Maintenance of the ModSAF baseline to include PTR Resolution
- d) Software Change Request (SCRs) Resolution
- e) User Support - may include providing help desk services to ModSAF users via Web pages and documentation
- f) Integrated Development Team and Change Control Board Support
- g) Adhoc ModSAF technical support to STRICOM when requested by the COR. This task will be provided on a Time and Materials basis. This task may include participation in ModSAF related working groups and conferences. This task may include keeping abreast of the work

19 June 1996

being performed by outside programs and assessing the benefits, impact, and cost of incorporating this work into the ModSAF baseline in coordination with STRICOM.

- h) Rearchitecture development of the ModSAF baseline to allow for component level testing and component level CM.
- i) ModSAF development to include the addition of new functionality and enhancements to existing functionality.

Appendix A

Report for DIS Testing of ModSAF 2.1 Beta

1. Introduction

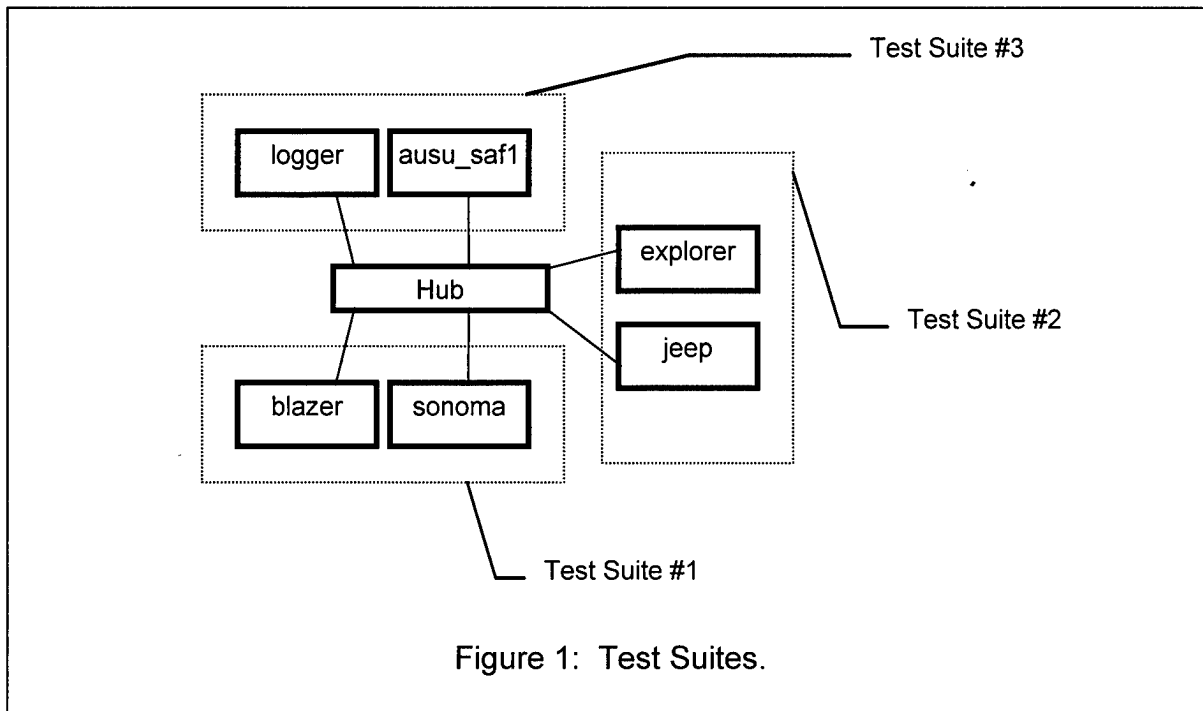
The purpose of this report is to document the testing of the ModSAF 2.1 Beta Version accomplished at the ADST-II Operational Support Facility (OSF). This report addresses: test objectives, test environment, test process, and findings.

1.1 OSF Objectives for ModSAF 2.1 Beta Test

- Determine ModSAF 2.1 Beta compliance with the DIS 2.04R Standard
- Evaluate the use of the DIS Test Suite (DTS) Version 3.2 (AcuSoft, Inc.) for testing the compliance of ModSAF Version 2.1 Beta against the DIS 2.04R Standard.
- Document the process of performing DIS Testing of ModSAF using the DTS software.

1.2 ModSAF 2.1 Beta Testing at the OSF

The testing activities were performed from 8 APR 96 to 19 APR 96 by representatives of Science Applications International Corporation (SAIC) and Simulation and Training Command (STRICOM) at the OSF located in Orlando, Florida. During this Beta Test, many sites concentrated on different areas of ModSAF to perform new capability tests and regression tests of the existing capabilities. The OSF was specifically responsible for performing DIS Testing which determines the compliance of ModSAF 2.1 Beta Version against the DIS 2.04R Standard. Only a few ModSAF entities were tested for DIS compliance to allow for: the evaluation of DTS software, and documenting the test process. Future DIS testing will strive toward compliance testing of all ModSAF entities.



2. Testing Environment

The host computers used for testing were configured to accommodate parallel testing of different aspects of the ModSAF software by multiple test engineers. Each test engineer utilizes a “test suite” for testing. The technical details of the testing environment are described below.

2.1 Lab Configuration

The configuration used for DIS testing of ModSAF 2.1 Beta is depicted by Figure 2, Table 1, and Table 2. A similar “test suite” and software configuration is suggested for future DIS testing of ModSAF, as the DTS software has heavy disk space requirements.

Host Name:	Platform:	IP Address:	Processor:	RAM:	Operating System:	Swap Space: 1 Blk = 512 bytes	Hard Disk:
logger	SGI Indy	164.217.53.30	MIPS R4400	96MB	IRIX 5.2	81880 Blks	1.0 GB
ausu_saf1	SGI Indy	164.217.53.10	MIPS R4400	96MB	IRIX 5.2	81880 Blks	1.0 GB
explorer	SGI Indy	164.217.53.199	MIPS R4400	128 MB	IRIX 5.3	81880 Blks	1.07 GB
jeep	SGI Indy	164.217.53.27	MIPS R4400	128 MB	IRIX 5.3	81880 Blks	1.07 GB
sonoma	SGI Indy	164.217.53.25	MIPS R4400	128 MB	IRIX 5.3	81880 Blks	1.07 GB
blazer	SGI Indy	164.217.53.201	MIPS R4400	128 MB	IRIX 5.3	81880 Blks	1.07 GB
Item:	Brand:	Model:	Miscellaneous:				
Hub/Repeater	Allied Telesis Inc.	Centre COM 3012T	Protocol: IEEE 802.3/Ethernet 10BaseT Capacity: 12 Ports				

Table 1: Hardware Configuration.

19 June 1996

Host Name:	Usage:	Version:	Options:
logger	N/A	N/A	N/A
ausu_saf1	N/A	N/A	N/A
explorer	ModSAF	2.1 Beta	-terrain itsec93-0101 -version 4 -disport 3001 -database 2
jeep	DTS	3.2	-disport 3001
sonoma	DTS	3.2	-disport 3000
blazer	ModSAF	2.1 Beta	-terrain itsec93-0101 -version 4 -disport 3000 -database 1

Table 2: Software Configuration.

2.3 Suggested Documentation

The following documentation provides the necessary reference information for performing DIS testing of ModSAF.

- DIS 2.04R Standard (IEEE1278.1 - 1995)
Provides DIS protocol and entity/munition enumeration information
- ModSAF 2.1 Beta Version Description Document (VDD)
Provides descriptions of new ModSAF capabilities pertinent to the release
- ModSAF 2.1 Beta User's Manual
Provides instructions for the operation of ModSAF software
- ModSAF 2.1 Beta Functional Description Document
Provides descriptions of existing ModSAF capabilities
- DIS Test Suite (DTS) User's Manual for Version 3.2
Provides instructions for the process and operation of DTS software

2.4 Test Personnel Qualifications

To effectively utilize the DTS software, individuals must have knowledge of: the DIS 2.04R Standard, the ModSAF source directory organization, and the layout and meaning of data file contents which describe entities. Typically, developers of ModSAF have this expertise.

3. DIS Testing Process

The DTS software dictates the testing process. As a result, the testing process may be divided into two phases: a preparation phase, and a ModSAF DIS testing phase. Most of the preparation phase may be accomplished prior to the acquisition of the latest ModSAF software; if existing models are to be tested then an existing version of ModSAF allows a head start. The activities performed by test personnel during the preparation phase are highly technical and labor intensive. Activities are less technical and labor intensive during the actual execution of the tests.

3.1 Preparation Phase

The preparation phase is comprised of the following activities: ModSAF entity selection, determining and documenting entity capabilities, recording entity capabilities, and generating test procedure statements. These activities are further described in detail below.

3.1.1 DTS Background

The DIS Test System provides a database of Capability Statement (CS) via the World Wide Web (WWW) at Universal Reference Locator (URL) <http://www.distest.org>. A CS is set up for each application to be tested for DIS compliance. The ModSAF application's CS has already been initiated into the database. All additional entries are modifications to the existing CS.

3.1.2 ModSAF Entity Selection

The first activity is to determine the ModSAF entities which will be tested. Due to the great number of supported ModSAF entities, this selection is usually a sample of the overall set. The criteria for selection is based upon the objectives of the tests. This criteria may dictate one vehicle per type or class.

3.1.3 Determining and Documenting Entity Capabilities

The goal of gathering the entity capabilities is to establish a baseline to test against. For each entity (any entity that transmits DIS Entity State PDUs) an entry into the ModSAF CS is needed. There are several sources available for gathering the information needed to complete the data entry of entity capabilities. ModSAF, through the editing of an instantiation of an entity, can show the weapon capabilities of an entity. The DTS provides a PDU scanner capability. Through this capability an Entity State PDU can be examined to gather information such as Entity Type, articulated parts, general appearances, and other. Other PDUs such as a Fire PDU can yield information pertaining to the weapon system's munitions such as munitions type, warhead type, fuse type, and others. ModSAF's *.rdr files are a prime resource for gathering capability information. In addition, Subject Matter Experts (SMEs) may be a valuable source of information.

During the entity capability gathering activity a few obstacles may be encountered. The level of control the ModSAF operator has over entities is limited. For example, when trying to gather information pertaining to the munitions of an entity, getting the entity to generate Fire PDUs is not straight-forward. The entity only fires when it reacts to a threat. Based on the threat, the entity decides the appropriate munitions to fire. Even using ModSAF's edit capability to zero out supply levels of all non-pertinent munitions does not ensure the entity will use the remaining munitions type. Effort and knowledge of ModSAF is needed to generate appropriate scenarios to generate PDUs of interest.

In general, to determine ModSAF capabilities for a given entity, perform the following from the home directory of the ModSAF installation:

1. `cd ~/common/src/ModSAF/entities`
2. view the file: `models.rdr`

This mapping file ties parameters to entities. Look for the entity of interest in the *models.rdr* file (i.e `vehicle_US_M1A2`).

3. Record the parameters associated with the entity

For the following entry,

“vehicle_US_M1A2” US_M1A2_MODEL_PARAMETERS GROUND_STD_PARAMS

the *US_M1A2_MODEL_PARAMETERS* & *GROUND_STD_PARAMS* are the parameters of interest.

4. Find the file in which the definition of each parameter¹ exists. This may be accomplished by using the grep command.

For example,

```
grep “GROUND_STD_PARAMS {“ *
```

yields the following output:

```
standard_params.rdr: GROUND_STD_PARAMS {
```

This indicates the file, *standard_params.rdr*, in which the definition for *GROUND_STD_PARAMS* occurs.

5. Peruse the definition of the parameters for information pertinent to the Capability Statement for the entity.
6. All significant data should be retained for subsequent completion of the entity’s capability statement.

3.1.4 Recording Entity Capabilities

The recording of entity capabilities is accomplished via the DIS Test Homepage. Using a WWW browser, open the DIS Test Homepage (at <http://www.distest.org>). At the homepage select the hypertext link (Press [here](#) on the DIS Test Homepage) to login. Login requires a user name and password. After entering the appropriate user name and password, press the submit query button. This will bring up the DIS Test Homepage Directory. From here select the hypertext link “Capability Statements” under the Applications Under Test Heading. The Capabilities Statement Menu will appear. Select the “*modify an existing Capability Statement*” link under “*Options for an Existing Capability Statement*.” At the Capability Statement Modify Menu select the ModSAF 2.1 capability statement and press the modify button. The CS for the application is displayed -- Press the “submit” button to access the CS. The CS is a menu driven process that will prompt the user for input. Enter all information gathered pertaining to the specific entity.

3.1.5 Generating Test Procedure Statements

¹ Note: There may be a maximum of three (3) parameter files used for each entity specification.

Based on the capabilities of the entity, a subset of tests within the DTS will be identified to test for DIS compliance. The tests to be performed are generated via this DIS Test Homepage. At the DIS Test Homepage Directory select the *Capability Statements* under the “*Applications Under Test*” section. From the Capability Statement Menu select the *Generate Test Procedure List* under the *Options for Existing Capability Statement* section. On the Capability Statement Test Procedure Menu select ModSAF 2.1 and press the *Generate Test Procedure List* button. This generates a DIS Test Procedure Profile for all entities within the application under test (i.e. ModSAF 2.1). This listing will identify all test procedures within the DTS that need to be performed to test for DIS compliance of the application. The profile is organized by: Entity Type, Test Procedure Paragraph, and Test Procedure Name.

3.2 ModSAF DIS Testing Phase

The ModSAF DIS Testing Phase utilizes the data from the preparation phase to semi-automate the testing process. The ModSAF DIS Testing phase is comprised of the following activities: software start-up, running applicable tests, performing data analysis, and documenting test results.

The basic configuration for a test suite consisted of two hosts computers. One host for the ModSAF 2.1 Beta application, and a second host for the DTS application. Multiple test suites may be simultaneously used to perform testing. This configuration allows parallel independent testing. If more than one test suite is to be concurrently operational, consideration must be given to ensure PO Databases and DIS ports are mutually exclusive. The DTS requires that the ITSEC'93 Terrain database be used.

3.2.1 Software Start-up

ModSAF is considered the Application Under Test (AUT) when performing DIS testing with the DTS.

The procedure for starting the ModSAF application is:

1. `cd ~/common/src/ModSAF/sgi_*`
2. `modsaf_sgi_* <options>`

The procedure for starting the DTS application is:

1. `cd ~/dts/dts-3.2`
2. `source ddtc (for C Shell)`
3. `entmgr <options>`

Once the DTS is started, the Program Info window can be closed by clicking on the OK button. At the DIS Test Suite's “Network Interface” window, select the menu option “Application Launcher” to launch the first three applications: Interactivity Tool Set, Test Executive, and Test Procedure Interface.

3.2.2 Running Applicable Tests

Once the Capability Statement has been completed/modified, you can obtain a list of Test Procedures to execute based on the information provided in the Capability Statement. This list can be generated by the DTS by selecting "Generate Test Procedure List" under the "Capability Statement" directory on the Home Page. This list will be used during testing to help identify which Test Procedures to execute.

The application's DIS capability information will also be used in the analysis portion of the testing process. This is accomplished by use of the Capability Profile which the DTS generates from the Capability Statement. The application's Capability Profile can be downloaded from the Home Page under "Generate Capability Profile" under the "Capability Statement" directory. Once the Capability Profile has been downloaded, it should be extracted from its tar format into the directory ~/dts-x.y/lib/dats (x.y refers to the DTS version number).

The tests identified by the Test Procedure List should be performed via the Test Procedure Interface Application. Any test not specified by the profile can be skipped by pressing the Skip button. On the Test Procedure Interface the AUT work area will provide the steps to be taken by the ModSAF operator. There may be the need for the ModSAF operator to improvise about how to best address the DTS instructions, as it is based upon certain assumptions (see the "Findings" section below for details). The DTS Instruction area will provide actions to be taken on the Interactivity Tool Set of the DTS. A test procedure is performed by the test personnel by alternately interacting with the DTS software and the ModSAF software, until all DTS instructions have been completed for a test procedure. This activity is repeated until all test procedures identified in the Test Procedure Profile have been completed. After all test procedures have been performed press the exit button on the Test procedure interface window.

The following information should be recorded in a test log book while performing the tests, for subsequent use during PDU analysis:

- Log file name for the test procedures
- Entity ID used on each test procedure

The log file name is important to ensure analysis is performed on the correct log file. The mapping of entity ID to the test procedure is needed since data analysis is performed on entity ID. It may take several iterations of analyzing the log file (one for each entity ID) to identify conformance/non-conformance to the DIS standard. This mapping is necessary since analysis of an existing log file is performed in respect to a single entity ID. Since several instantiation of an entity may be used, to complete all test procedures for an entity type, multiple analysis of the log file may be necessary. Also note, the Test Procedure List and Capability Statement Profile are for all entity types in ModSAF -- Each entity type will have a separate entity ID.

For additional information on the use of the Test Executive Tool refer to the DIS Test Suite (DTS) User's Manual.

3.2.3 Performing Data Analysis

The DTS provides a Data Analysis Tool. This tool is used to analyze the log file recorded during the performance of test procedures. Launch the Data Analysis Tool from the DIS Test Suite's "Network Interface" via the menu option "Application Launcher." On the Data Analysis Tool select the menu

option "File" and choose the option "open AUT." Enter the name of the log file for the application under test. The tool will provide an analysis of the PDU logger file. Test personnel enter information from the test log book as required during the analysis process. The analysis will identify areas where the application has violated the DIS standard. Test Personnel may have to verify that the results are accurate by interpreting how the conclusions were derived from the PDU log. This is accomplished by manually examining PDU information, using the Data Analysis Tool, against the DIS 2.04R Standard.

For additional information on the use of the Data Analysis Tool refer to the DIS Test Suite (DTS) User's Manual.

3.2.4 Documenting Test Results

All information gathered pertaining to the capabilities of an entity, either recorded during the test procedure or obtained through data analysis, should be input into the final report on entity compliance; this includes any interpretations of the analysis provided by test personnel. All deviations from the standard should map to a Problem/Trouble Report (PTR).

The following data should be retained for future testing and organized by entity:

- Capability Statement
- Capability Profile
- Test Procedure Profile
- PDU Logger File
- Test Instruction Log
- Test Analysis Log
- Test Log Book Entries
- Associated PTRs

This information will support the testing of associated PTR fixes, as well contribute to a suite of regression tests.

4. Findings

The following findings are based upon DIS testing using DTS 3.2 software and ModSAF 2.1 Beta software for the following entities: M1A2, T72M, AH-64, Mi24, Dismounted Infantry (US), Dismounted Infantry (USSR). Fixed-wing aircraft testing was planned, but not accomplished due to existing PTRs with these aircraft.

1. DTS is based upon DIS enumerations, dated March 1995. The DTS software indicates that ModSAF entities fail some tests due to Enumerations used by the ModSAF application which are not recognized by the DTS software. This fact requires test personnel to manually verify the results for those test procedures that "failed" to determine if it was actually due to the enumeration values. If the "failure" was attributed to an enumeration, the subject test procedure was considered to have passed and the unrecognized enumeration is noted in the final test results.

2. DTS provides limited support for testing the ModSAF application using all terrain databases. Currently only the "itsec93-0101" database is supported by the DTS software.
3. The DTS Test Procedures are oriented toward the testing of manned simulators, in that they assume the test personnel have detailed control over the entity's characteristics and capabilities. Note that fidelity levels for a ModSAF entity differs from that of a manned simulator for the same entity. Manned simulators typically have actuators which are consistent with the real world entity, or "trainer unique devices" which allow specific control over its DIS communicated existence in a simulation.

The following items represent examples which support this finding:

- The DTS software expects that the entity is capable of firing at a particular terrain location which is not occupied by an opposing entity.
 - Firing one of each weapon/munitions is a single test procedure step, which requires many non-DTS instructed steps for operating the ModSAF software to introduce adequate opposition entities which allow the ModSAF entity under test to fire each intended weapon/munitions. This requires the test personnel to have a technical understanding of the developed behavior of the ModSAF entity which is usually afforded only by: the developer of the behaviors for the particular entity, or a Subject Matter Expert.
 - Although ModSAF does support radio communications, it does not support it to the fidelity required by the DTS software. A particular test procedure requires the "keying" of the radio hand-set to pass the test.
 - All stances for Dismounted Infantry cannot be easily executed.
 - Turret scanning for a full 360 degree sweep cannot be accomplished since ModSAF behaviors are not programmed to allow this fidelity. A ModSAF entity's turret scan sector is typically limited by its role in a unit.
4. Completing a Capability Statement on the WWW at times requires a technical interpretation of the question, to discern the appropriate answer based upon detailed knowledge of the ModSAF software (see "radio fidelity" bullet in the Findings section above as an example).
 5. Completing a Capability Statement per entity is labor intensive. It would be preferable to have a Capability Statement which allows similar entities to share test procedures, as ModSAF is likely in the future to have hundreds of supported entities.
 6. The testing of each entity requires the following computer data files which consume disk space: a capability profile, and a PDU logger file. Additional DTS files/logs should be retained for effective PTR testing or regression testing. Much disk space is required to retain adequate data for all ModSAF supported entities.
 7. The DTS software assumes that the same entity ID is used for all test procedures in a test class (e.g. Dead Reckoning Algorithm tests). A manual mapping of entity ids is necessary when analyzing the PDU logger file. During PDU analysis, multiple entity ids are required at times

(e.g. one entity ID for Test Procedure 2.4.4.2 and another for Test Procedure 2.4.6.2). This need arises when an existing ModSAF entity under test is damaged, as a result of the test procedure steps, to the extent that it prevents further use of the entity.

5. Conclusions

- Although the DTS 3.2 software is apparently excellent for testing the DIS compliance of manned simulators, it is not currently adequate for the testing of ModSAF software or other Computer Generated Forces (CGF); based upon the above reported findings. DTS 3.2 software will require enhancements to become more suitable for testing ModSAF for DIS compliance. This may be accomplished over time with cooperation between group members from STRICOM, AcuSoft Inc., and ModSAF.
- While some test procedures could not be performed as written, the DTS was still able to flag several DIS requirements that were not being met by ModSAF. These requirements, including dead reckoning, incorrect issuance of transmitter PDUs, not incrementing the "event" field in the emissions PDU, and incorrect enumerations were identified and documented as PTRs.
- This report captures a testing process for using the DTS software to test ModSAF software for DIS Compliance. It also represents an overview for test personnel that may not be familiar with the use of the DTS product. The testing process could be enhanced in the future to contain additional details.

Appendix B

Pricing Strategy

ModSAF Software Development

Pricing Strategy

The following figure lists the task categories that will be considered as a basis for developing rough order of magnitude (ROM) costs for ModSAF development activities.

19 June 1996

ModSAF Software Development Pricing Strategy	Requirements Analysis	Design Analysis	Knowledge Acquisition	Knowledge Engineering	Coding	Test & Debug	Documentation	Baseline Integration	VV&A Activities	
Software Engineering										
Entities										
Physical Models										
Behavioral Models										
Data File Modifications										
Architectural Extensions										
PO Database Modifications.										
Terrain Database Modifications.										
Simulation Protocol Extensions										
Other Extensions										
Engineering Support										
Site Support										
Experiment Design Support										
Experiment Execution Support										
Customer Demonstration										
Meeting Support										
Standards Committee Meetings										
ModSAF Installation Support										
Travel										
Media Preparation & Delivery										
Hardware & Software Deliverables										
Project Support										
Preliminary Work Statement & ROM										
Proposal Preparation										
CDRLs										
ADST II Monthly Program Reviews										
Program Management										
Program Operations Support										
Contracts Support										
Quality Assurance										
Configuration Management										
Subcontracts Management										
Scheduling										

Each of the engineering tasks may require more than one instance, and a table entry will be created for each. As an example, a SOW may require three physical models, and the table would therefore contain three separate estimates for physical modeling. Each of the engineering tasks will be evaluated against each of the categories listed across the top of the table, and additionally against other factors that are specific to each type of task. These other factors are:

The following will be considered for each Entity:

- How complex is the entity?
- Must the entity be created from scratch, or can it be implemented as a modification to an existing entity?
- Can the entity be created using existing physical components, or will it require the implementation of additional physical models such as weapons, sensors or vehicle dynamics?

The following will be considered for each Physical Model:

- How complex is the model?
- Will an existing model be modified, or is a new model required?
- If a new model is required, is an existing model available (coded), or must the model be written from scratch?
- If a coded model is available, does it have sufficient performance for real time execution within the ModSAF context.
- If a coded model is available, is a language conversion necessary?
- If a coded model is not available, will model algorithms be provided, or will research be required to determine suitable algorithms?

The following will be considered for each Behavior:

- How complex is the behavior?
- Will an existing behavior be modified, or is a new behavior required?
- If a new behavior is required, how many states will the behavior require?

The following will be considered for each Data File Requirement:

- Will existing data files be modified, or will new data files be added?
- Is the format of new data files currently supported within ModSAF?

- If the new data file format is not supported, will code be written to support the format, or will the data file be converted to a supported format?
- If a data file format conversion is required, will a conversion tool be included as part of the ModSAF baseline?

The following will be considered for each Architectural Modification:

- Architectural modifications can vary greatly in scope and risk, and for this reason such modifications will be estimated on a case by case basis by engineers with extensive knowledge and experience in ModSAF architectural design.

The following will be considered for each Persistent Object Database Extension:

- How complex is the PO Database object?

The following will be considered for each Terrain Database Modification:

- Can the new functionality be implemented as a correction to an existing ModSAF terrain database?
- Can the required database be compiled from an existing database in another format such as S1000?
- If the database must be created from scratch, are all of the source materials available?

Other ModSAF modifications will be considered on a case by case basis.

Engineering support tasks will be estimated as required by the specifics of the program, and program support tasks will generally be estimated as a percentage of the total engineering costs. A spread sheet has been developed that automates and standardizes this ROM costing methodology by applying numerical values and formulas to the tasks described above. These values and formulas, as well as the methodology itself, may be changed as experience is gained with ADST II programs in an effort to continuously improve ROM costing accuracy.